STOCK MARKET PREDICTION

MINI PROJECT REPORT

Submitted by

GOKULANAND P (202209013)
ARUNKUMAR K (202209006)
KANAGABALA R (202209023)

in

19AD451 – DATA ANALYTICS LABORATORY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

MEPCO SCHLENK ENGINEERING COLLEGE
SIVAKASI

MAY 2024

MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI

AUTONOMOUS

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

BONAFIDE CERTIFICATE

This is to certify that it is the bonafide work of **“GOKULANAND.P (202209013), ARUNKUMAR.K (202209006), KANAGABALA.R (202209023)”** for the mini project titled **“STOCK MARKET PREDICTION”** in 19AD451–Data Analytics Laboratory during the fourth semester January 2024 – June 2024 under my supervision.

SIGNATURE	**SIGNATURE**
Dr.R.Nirmalan,	**Dr. J. Angela Jennifa Sujana,**
Assistant Professor,	**Professor & Head**,
AI&DS Department,	AI&DS Department
Mepco Schlenk Engg., College,	Mepco Schlenk Engg., College,
Sivakasi - 626 005.	Sivakasi - 626 005

ACKNOWLEDGEMENT

I sincerely thank our respected principal **Dr.S.Arivazhagan,** for providing necessary facilities to carry out this work successfully.

I wish to express my sincere gratitude to **Dr.J.Angela Jennifa Sujana** Professor and Head of the Artificial Intelligence & Data Science department for her stimulating support and encouragement for the completion of this project work.

I am grateful to our Project guides and **Dr.R.Nirmalan,** Assistant Professor and **Dr.P.Thendral**, Associate Professor.

With deep sense of gratitude, I would like to thank our Head of the Artificial Intelligence & Data Science department for her insightful comments and valuable suggestions which helped me to complete this project work successfully.

My sincere thanks to our revered **faculty members** and **lab technicians** for their help over this project work.

Last but not the least, I extend my indebtedness towards my **beloved family and friends** for their support which made this project a successful one.

ABSTRACT

The stock market is a complex and dynamic system influenced by a multitude of factors, making accurate prediction a challenging yet valuable endeavor. This study explores the application of linear regression, a fundamental machine learning technique, for predicting stock market prices. By leveraging historical stock data, including opening, closing, high, and low prices, along with trading volume, we aim to construct a predictive model that can forecast future stock prices. The simplicity and interpretability of linear regression make it an attractive method for this purpose, providing insights into the relationships between different market variables.

Our approach involves preprocessing the data to handle missing values and normalize features, followed by the implementation of a linear regression model. We evaluate the model's performance using key metrics such as Mean Squared Error (MSE) and R-squared (R^2) to assess its accuracy and robustness. The results demonstrate that while linear regression offers a straightforward method for stock price prediction, its effectiveness is subject to the linearity assumption and may be limited in capturing the market's inherent volatility and non-linear trends.

This study underscores the potential and limitations of linear regression in stock market prediction, suggesting that while it can serve as a useful baseline model, more sophisticated techniques, possibly incorporating machine learning and deep learning algorithms, may be required for enhanced predictive performance. The findings contribute to the ongoing research in financial forecasting, providing a foundational understanding for future explorations into more advanced predictive models.

TABLE OF CONTENTS

CHAPTER 1

1 INTRODUCTION

1.1 Overview of Project

Predicting stock market movements is a complex task that involves various factors such as economic indicators, company performance, geopolitical events, and investor sentiment. One approach to forecast stock prices is using linear regression, a statistical technique that models the relationship between independent variables and a dependent variable.

In the context of stock market prediction, linear regression aims to identify patterns and trends in historical stock price data and other relevant factors to predict future price movements. The independent variables could include factors like company earnings, market indices, interest rates, and industry trends, while the dependent variable is typically the stock price.

To apply linear regression for stock market prediction, historical data is collected and divided into training and testing sets. The training set is used to fit the regression model by estimating the coefficients that best describe the relationship between the independent and dependent variables. Once the model is trained, the testing set is used to evaluate its performance and assess its predictive accuracy.

However, it's important to note that linear regression assumes a linear relationship between the independent and dependent variables, which may not always hold true in the dynamic and complex stock market environment. Additionally, stock prices are influenced by numerous unpredictable factors, making accurate prediction challenging

1.2 Technology Used

The project utilizes several technologies and tools to achieve its objectives. These include:

1.2.1 Data Collection and Storage

- **APIs for Financial Data**

Services like Alpha Vantage, Yahoo Finance API, and Quandl provide historical stock prices and trading volume data.

1.2.2 Data Preprocessing

- **Python**

A popular programming language for data science, equipped with libraries such as Pandas and NumPy for data manipulation and cleaning.

- **Jupyter Notebooks**

An interactive computing environment for data analysis and visualization.

1.2.3 Exploratory Data Analysis (EDA)

- **Yfinace**

Yahoo! Finance is a media property that is part of the Yahoo! network. It provides financial news, data and commentary including stock quotes, press releases, financial reports, and original content. It also offers some online tools for personal finance management. In addition to posting paid partner content from other web sites, it posts original stories by its team of staff journalists. It is ranked 20th by SimilarWeb on the list of largest news and media websites

- **Matplotlib and Seaborn:**

Python libraries for creating static, animated, and interactive visualizations to understand data patterns and relationships.

1.2.4 Model Development

- **Scikit-learn**

A robust machine learning library in Python that provides tools for linear regression and model evaluation.

- **Statsmodels**

Another Python library that offers more advanced statistical models and hypothesis tests.

1.2.5 Linear Regresssion

- The very simplest case of a single scalar predictor variable *x* and a single scalar response variable *y* is known as simple linear regression. The extension to multiple and/or vector -valued predictor variables (denoted with a capital *X*) is known as **multiple linear regression**, also known as **multivariable linear regression** (not to be confused with multivariate linear regression.

  Multiple linear regression is a generalization of simple linear regression to the case of more than one independent variable, and a special case of general linear models, restricted to one dependent variable. The basic model for multiple linear regression is

$$Y_{ij} = \beta_{0j} + \beta_{1j}X_{i1} + \beta_{2j}X_{i2} + \ldots + \beta_{pj}X_{ip} + \epsilon_{ij}$$

- For each observation In the formula above we consider *n* observations of one dependent variable and *p* variables.

1.2.5 Model Evaluation

- **Metrics:**

Mean Squared Error (MSE) and R-squared (R^2) are computed using Scikit-learn to assess the model's performance.

1.2.6 Deployment

- **Flask or Django**

Web frameworks in Python for developing web applications that serve the predictive model.

- **Docker**

For containerizing the application, ensuring consistent environments across different deployment stages.

- **Cloud Platforms**

Services like AWS, Google Cloud, or Azure for scalable and reliable hosting of the application.

CHAPTER 2

2.1 LITERATURE REVIEW

The prediction of stock market prices has been a subject of extensive research in financial economics and machine learning due to its potential for high financial rewards. Various methodologies, including statistical models and machine learning algorithms, have been explored to predict market trends and prices. This literature review summarizes key contributions in the field, particularly focusing on the use of linear regression for stock market prediction.

2.1.1 Fundamental Analysis

Fundamental analysis involves evaluating a stock's intrinsic value by examining related economic, financial, and other qualitative and quantitative factors. Graham and Dodd's seminal work "Security Analysis" laid the foundation for fundamental analysis by focusing on financial statements, earnings, dividends, and growth potential to determine stock value . However, this approach is often criticized for being less responsive to market sentiments and technical indicators.

2.1.2 Technical Analysis

Technical analysis, as discussed in Murphy's "Technical Analysis of the Financial Markets," relies on historical price and volume data to predict future price movements. It operates on the premise that market prices reflect all available information and that prices move in trends that can be identified and exploited . This approach is highly data-driven and forms the basis for many algorithmic trading strategies.

2.1.3 Statistical Methods

Linear regression, a simple yet powerful statistical method, has been widely used for predicting stock prices. In their study, Fama and French utilized linear regression to explore the relationship between stock returns and various economic factors, developing their famous three-factor model which explains much of the variability in stock returns . Despite its simplicity, linear regression's assumption of linearity can be a significant limitation in capturing the complex, non-linear nature of financial .

2.1.4 Machine Learning Approaches

The advent of machine learning has introduced more sophisticated methods for stock market prediction. Guresen et al. compared the performance of neural networks, linear regression, and support vector machines in predicting stock prices. Their findings indicated that while neural networks often outperform linear regression due to their ability to model non-linear relationships, linear regression remains valuable for its simplicity and interpretability .

2.1.5 Hybrid Models

Recent research has focused on hybrid models that combine linear regression with other techniques to improve prediction accuracy. For instance, Kim and Han proposed a hybrid approach that integrates genetic algorithms with linear regression to optimize feature selection and improve prediction performance . Such hybrid models leverage the strengths of multiple methods to address the inherent limitations of individual approaches.

2.1.6 Evaluation of Linear Regression

In their comprehensive study, Patel et al. evaluated the efficacy of various machine learning models, including linear regression, for stock market prediction. They highlighted that linear regression, while not the most accurate, serves as a useful benchmark for comparing more complex models . Its straightforward implementation and ease of interpretation make it an essential tool for initial model development and baseline comparison.

2.2 Real-Time Applications

2.2.1 Automated Trading Systems

Automated trading systems, also known as algorithmic trading, utilize real-time stock market predictions to execute trades based on predefined criteria. By integrating a linear regression model, these systems can predict short-term price movements and automatically place buy or sell orders to capitalize on these predictions. The use of real-time data ensures that the system reacts promptly to market changes, optimizing trade execution and potentially increasing profitability. The speed and accuracy provided by real-time predictions are crucial for high-frequency trading strategies where milliseconds can make a significant difference.

2.2.2 Investment Advisory Services

Investment advisory services can leverage real-time stock market predictions to provide clients with timely and informed advice. By incorporating a linear regression model into their analysis tools, advisors can offer data-driven recommendations based on the latest market conditions. This allows them to suggest optimal entry and exit points for investments, tailor strategies to individual client goals, and enhance the overall decision-making process. Real-time insights enable advisors to respond quickly to market shifts, thereby improving the quality and relevance of their advice.

2.2.3 Risk Management

Real-time stock market predictions are invaluable for risk management purposes. Financial institutions and investment firms can use linear regression models to forecast potential price movements and identify risks in their portfolios. By continuously monitoring real-time data, these institutions can anticipate adverse market trends and take proactive measures to mitigate losses. This might involve rebalancing portfolios, adjusting hedging strategies, or setting stop-loss orders. The ability to predict and respond to market changes in real time enhances the overall risk management framework, ensuring better protection against market volatility.

2.2.4 Market Analysis and Research

Market analysts and researchers can benefit from real-time stock market predictions by gaining insights into market dynamics and trends. A linear regression model can process real-time data to uncover patterns and correlations that might not be apparent through traditional analysis. Researchers can use these insights to publish timely reports, develop market strategies, and provide forecasts that aid in strategic planning. Real-time analysis tools empower analysts to stay ahead of market developments and offer more accurate and current assessments of market conditions.

2.2.5 Personal Investment Tools

Individual investors can utilize real-time stock market prediction tools to make more informed decisions. Apps and online platforms can integrate linear regression models to provide users with predictions on stock price movements, personalized investment recommendations, and alerts on significant market changes. These tools help investors manage their portfolios more effectively, identify opportunities, and minimize risks. Real-time updates ensure that users have the latest information at their fingertips.

CHAPTER 3

3.1 PROPOSED SYSTEM

To build an effective real-time stock market prediction system using linear regression, the system architecture needs to seamlessly integrate data collection, preprocessing, model training, prediction, and result dissemination. Here's a detailed breakdown of the proposed system:

3.1.1 Data Collection and Integration

The proposed system begins with robust data collection and integration. Real-time stock data from reliable sources such as Alpha Vantage, Yahoo Finance, and Quandl is continuously fetched using APIs. This includes historical data, real-time stock prices, trading volumes, and other relevant financial indicators. The collected data is stored in a scalable and structured database, such as PostgreSQL for relational data, and MongoDB for semi-structured or unstructured data. The system ensures that data is updated in real time, providing a constant stream of the latest market information for analysis.

3.1.2 Data Preprocessing and Feature Engineering

Once the data is collected, it undergoes preprocessing to ensure quality and consistency. This involves cleaning the data by handling missing values, removing outliers, and correcting any inconsistencies. Feature engineering is then applied to create new variables that enhance the predictive power of the model. Key features include moving averages, relative strength index (RSI), volatility indicators, and other technical and financial metrics. Normalization techniques are applied to scale the data, making it suitable for linear regression modeling. The preprocessing step is crucial for transforming raw data into meaningful inputs for the model.

3.1.3 Model Training and Evaluation

With a clean and feature-rich dataset, the system trains a linear regression model to predict stock prices. The data is split into training and testing sets to evaluate the model's performance. The linear regression algorithm is implemented using libraries such as Scikit-learn, which provides tools for model training, evaluation, and optimization. Key performance metrics such as Mean Squared Error (MSE) and R-squared (R^2) are used to assess the model's accuracy and fit. The evaluation process ensures that the model generalizes well to new, unseen data and provides reliable predictions.

3.1.4 Real-Time Prediction Engine

The core of the proposed system is the real-time prediction engine. Using Apache Kafka for real-time data streaming, the system processes incoming stock market data continuously. The trained linear regression model is applied to this real-time data to generate immediate predictions of stock prices. The prediction engine is designed to handle data in mini-batches to balance computational efficiency and prediction latency. This component ensures that users receive timely and accurate stock price forecasts, enabling them to make informed trading decisions in real time.

3.1.5 User Interface and API

To make the system accessible, a user-friendly interface and API are developed. A web application built using Flask or Django displays real-time predictions and interactive visualizations of stock data and trends. Users can access the predictions through an intuitive dashboard, view historical performance, and set up alerts for significant market movements. Additionally, a RESTful API provides programmatic access to the predictions, allowing third-party applications and services to integrate the system's capabilities. This interface ensures that both individual investors and institutional users can benefit from the real-time predictive insights.

3.2 PROPOSED SOLUTION

3.2.1 Comprehensive Data Collection

The proposed solution starts with a comprehensive data collection strategy. This involves gathering real-time and historical stock market data from multiple reliable sources, such as Alpha Vantage, Yahoo Finance, and Quandl. Data points include opening prices, closing prices, high and low prices, trading volumes, and other financial indicators. By leveraging robust APIs, the system ensures continuous and real-time data acquisition. This data is stored in a highly scalable database system, enabling efficient access and management of vast amounts of information necessary for accurate stock market predictions.

3.2.2 Advanced Data Preprocessing

Once the data is collected, it undergoes advanced preprocessing to prepare it for modeling. This involves cleaning the data by addressing missing values, removing outliers, and standardizing formats. Feature engineering is a critical step where new features such as moving averages (e.g., 5-day, 20-day), technical indicators (e.g., RSI, MACD), and volatility measures (e.g., ATR) are created. These features enhance the predictive power of the model. The data is then normalized to ensure all features are on a similar scale, which is essential for the effectiveness of the linear regression model.

3.2.3 Robust Model Training

With a clean and feature-rich dataset, the next step is robust model training. A linear regression model is trained using historical stock data to predict future stock prices. The dataset is split into training and testing sets to evaluate the model's performance objectively. Using Python libraries like Scikit-learn, the model is trained and its parameters are optimized to achieve the best possible predictive accuracy. Key metrics such as Mean Squared Error (MSE) and R-squared (R^2) are used to assess the model's performance. The model is tuned to ensure it generalizes well to new data, providing reliable predictions.

3.2.4 Real-Time Prediction Framework

The core of the proposed solution is a real-time prediction framework. This framework uses Apache Kafka for real-time data streaming, ensuring the system processes incoming stock market data with minimal latency. The trained linear regression model is deployed to make real-time predictions as new data arrives. This setup allows for immediate forecasting of stock prices, enabling users to act on the most current market information. The prediction framework is designed to be highly efficient, handling large volumes of data in real time while maintaining high accuracy.

3.4.5 User-Friendly Interface and API

To make the system accessible, a user-friendly interface and API are integral parts of the solution. A web application, developed using frameworks such as Flask or Django, provides users with an interactive dashboard to view real-time predictions, historical data trends, and visualizations. Users can customize their experience by setting up alerts for specific stock movements and accessing detailed reports.

CHAPTER 4

4.1 DATASET

4.1.1 SBIN.NS (State Bank of India)

- **Industry:** Banking and Financial Services
- **Explanation:** State Bank of India (SBI) is a multinational, public sector banking and financial services statutory body. It is the largest bank in India, providing a wide range of banking products and services to individuals and businesses.
- **Relevance:** As the largest bank in India, SBI plays a crucial role in the country's economy. It has a significant influence on financial stability and economic development in India.

4.1.2 RELIANCE.NS (Reliance Industries Limited)

- Industry: Conglomerate (Energy, Petrochemicals, Textiles, Natural Resources, Retail, Telecommunications)
- **Explanation:** Reliance Industries Limited (RIL) is one of the largest conglomerates in India, with a diverse range of businesses including energy, petrochemicals, textiles, natural resources, retail, and telecommunications.
- **Relevance:** RIL is one of the most valuable companies in India, significantly impacting the economy. Its ventures into telecommunications (Reliance Jio) and retail have transformed these sectors.

4.1.3 TCS.NS (Tata Consultancy Services)

- **Industry:** Information Technology and Services
- **Explanation**: Tata Consultancy Services (TCS) is a leading global IT services, consulting, and business solutions organization. It is a part of the Tata Group.
- **Relevance**: TCS is one of the largest IT services firms globally, contributing significantly to the IT industry and the Indian economy. It is also a major player in digital transformation services worldwide.

4.1.4 HDFC.NS (Housing Development Finance Corporation)

- **Industry:** Banking and Financial Services

- **Explanation:** HDFC is a leading provider of housing finance in India. It also has a significant presence in banking, life and general insurance, asset management, venture capital, realty, and education loans.
- **Relevance:** HDFC has played a pivotal role in providing housing finance and has been instrumental in shaping the housing finance sector in India. Its banking subsidiary, HDFC Bank, is one of the largest and most valued private sector banks in India.

4.1.5 INFY.NS (Infosys Limited)

- **Industry:** Information Technology and Services
- **Explanation:** Infosys is a global leader in next-generation digital services and consulting. It provides software development, maintenance, and independent validation services to companies in finance, insurance, manufacturing, and other domains.
- **Relevance**: Infosys is a key player in the global IT services industry and a major contributor to India's IT boom. It is known for its innovative solutions and extensive R&D efforts.

4.1.6 ICICIBANK.NS (ICICI Bank Limited)

- **Industry**: Banking and Financial Services
- **Explanation:** ICICI Bank is one of the largest private sector banks in India, offering a wide range of banking products and financial services to corporate and retail customers.
- **Relevance:** ICICI Bank is a significant player in the Indian banking sector, known for its innovation in banking products and digital banking initiatives.

4.1.7 TATAMOTORS.NS (Tata Motors Limited)

- **Industry:** Automotive
- **Explanation:** Tata Motors is a leading global automobile manufacturer of cars, utility vehicles, buses, trucks, and defense vehicles. It is a part of the Tata Group.
- **Relevance:** Tata Motors is a major player in the Indian automotive industry and has a significant presence in international markets. Its acquisition of Jaguar Land Rover has further enhanced its global footprint.

4.1.8 HINDUNILVR.NS (Hindustan Unilever Limited)

- **Industry:** Consumer Goods
- **Explanation**: Hindustan Unilever Limited (HUL) is a leading fast-moving consumer goods

(FMCG) company in India. It has a wide portfolio of products including foods, beverages, cleaning agents, personal care products, and water purifiers.

- **Relevance:** HUL is one of the largest and most well-known FMCG companies in India, with a vast distribution network and strong brand presence. It plays a crucial role in the daily lives of millions of consumers.

4.1.9 MARUTI.NS (Maruti Suzuki India Limited)

- **Industry**: Automotive
- **Explanation:** Maruti Suzuki is a subsidiary of the Japanese automaker Suzuki Motor Corporation and is the largest automobile manufacturer in India. It produces a wide range of cars from affordable small cars to premium SUVs.
- **Relevance:** Maruti Suzuki dominates the Indian car market with a significant market share, making it a key player in the automotive sector. Its products are known for reliability and affordability.

4.1.10 BAJFINANCE.NS (Bajaj Finance Limited)

- **Industry**: Financial Services
- **Explanation:** Bajaj Finance is a subsidiary of Bajaj Finserv and one of the leading non-banking financial companies (NBFCs) in India. It offers a variety of financial products and services including consumer finance, SME finance, commercial lending, and wealth management.
- **Relevance:** Bajaj Finance is a prominent player in the Indian financial services sector, known for its innovative products and strong customer focus. It has a significant influence on the consumer finance market in India.

4.2 FEATURES IDENTIFICATION

In stock market prediction, feature identification is crucial to enhance the model's predictive performance. Effective feature selection can lead to a more accurate and interpretable linear regression model. Here are the key points to consider when identifying features for predicting stock prices:

4.2.1 Historical Price Data

- **Open Price:** The price at which a stock opens at the start of the trading day.
- **Close Price:** The price at which a stock closes at the end of the trading day.
- **High Price:** The highest price a stock reaches during the trading day.
- **Low Price:** The lowest price a stock reaches during the trading day.
- **Volume:** The total number of shares traded during the trading day.
- **Adjusted Close Price:** The closing price adjusted for corporate actions like dividends and stock splits.

4.2.2 Moving Averages

- **Simple Moving Average (SMA)**

Average stock price over a specific number of days. Common periods are 5-day, 20-day, 50-day, and 200-day SMAs.

- **Exponential Moving Average (EMA)**

A type of moving average that gives more weight to recent prices, thus reacting more quickly to price changes.

4.2.3 Volatility Measures

- **Average True Range (ATR)**

Measures market volatility by decomposing the entire range of an asset price for a given period, typically 14 days.

- **Volatility Index (VIX)**

Often referred to as the "fear index," it represents market expectations of near-term volatility.

4.2.4 Economic Indicators

- **Interest Rates**

Changes in interest rates can influence stock prices, as they affect borrowing costs and consumer spending.

- **Inflation Rates**

High inflation can erode purchasing power and impact stock market performance.

- **Gross Domestic Product (GDP)**

A measure of a country's economic health, with growth indicating a strong economy which can positively affect stock prices.

4.2.5 Company-Specific Financial Metrics

- **Earnings Per Share (EPS)**

  Indicator of a company's profitability. Higher EPS often correlates with higher stock prices.

- **Price-to-Earnings Ratio (P/E Ratio)**

  Valuation ratio comparing a company's current share price to its per-share earnings.

- **Dividend Yield**

  A financial ratio that shows how much a company pays out in dividends each year relative to its stock price.

CHAPTER 5

5.1 PROGRAM CODE

The coding part of this project involves the implementation of the predictive model using Python and Streamlit for the user interface. Here's a breakdown of the key components and steps involved in the coding part:

```
import yfinance as yf
import streamlit as st
import plotly.express as px
import pandas as pd
from sklearn.linear_model import LinearRegression

st.title("Stock Data and Price Prediction")

tickers = ['SBIN.NS', 'RELIANCE.NS',
'TCS.NS','HDFC.NS','INFY.NS','ICICIBANK.NS','TATAMOTORS.NS','HINDUNILVR.NS','MAR
UTI.NS','BAJFINANCE.NS']

selected_ticker1 = st.selectbox("Choose the first stock ticker:", tickers)

selected_ticker2 = st.selectbox("Choose the second stock ticker:", tickers)

stock1 = yf.Ticker(selected_ticker1)
stock_info1 = stock1.info

current_price1 = stock_info1.get('currentPrice', 'N/A')
day_high1 = stock_info1.get('dayHigh', 'N/A')
day_low1 = stock_info1.get('dayLow', 'N/A')
```

```
previous_close1 = stock_info1.get('previousClose', 'N/A')
volume1=stock_info1.get('volume','N/A')

stock2 = yf.Ticker(selected_ticker2)
stock_info2 = stock2.info

current_price2 = stock_info2.get('currentPrice', 'N/A')
day_high2 = stock_info2.get('dayHigh', 'N/A')
day_low2 = stock_info2.get('dayLow', 'N/A')
previous_close2 = stock_info2.get('previousClose', 'N/A')
volume2=stock_info2.get('volume','N/A')

st.subheader("Current Stock Data")
col1, col2 = st.columns(2)

with col1:
    st.write(f"**Current Price of {selected_ticker1}:** {current_price1}")
    st.write(f"**Day High:** {day_high1}")
    st.write(f"**Day Low:** {day_low1}")
    st.write(f"**Previous Close:** {previous_close1}")
    st.write(f"**Volume :** {volume1}")

with col2:
    st.write(f"**Current Price of {selected_ticker2}:** {current_price2}")
    st.write(f"**Day High:** {day_high2}")
    st.write(f"**Day Low:** {day_low2}")
    st.write(f"**Previous Close:** {previous_close2}")
    st.write(f"**Volume :** {volume2}")

stock_hist1 = stock1.history(period="5y")
stock_hist2 = stock2.history(period="5y")
```

```
st.subheader("Historical Data for the Last 5 years")
col3, col4 = st.columns(2)

with col3:
    st.write(f"**{selected_ticker1} Historical Data**")
    st.write(stock_hist1)

with col4:
    st.write(f"**{selected_ticker2} Historical Data**")
    st.write(stock_hist2)

fig1 = px.line(stock_hist1, x=stock_hist1.index, y='Close', title=f'{selected_ticker1} Stock Price - Last 5 years')
st.plotly_chart(fig1)

fig2 = px.line(stock_hist2, x=stock_hist2.index, y='Close', title=f'{selected_ticker2} Stock Price - Last 5 years')
st.plotly_chart(fig2)

fig_open1 = px.line(stock_hist1, x=stock_hist1.index, y='Open', title=f'{selected_ticker1} Opening Prices - Last  5 years')
st.plotly_chart(fig_open1)

fig_open2 = px.line(stock_hist2, x=stock_hist2.index, y='Open', title=f'{selected_ticker2} Opening Prices - Last  5 years')
st.plotly_chart(fig_open2)

fig_high1 = px.line(stock_hist1, x=stock_hist1.index, y='High', title=f'{selected_ticker1} High Prices - Last 5 years')
st.plotly_chart(fig_high1)

fig_high2 = px.line(stock_hist2, x=stock_hist2.index, y='High', title=f'{selected_ticker2} High
```

```
Prices - Last 5 years')
st.plotly_chart(fig_high2)

fig_low1 = px.line(stock_hist1, x=stock_hist1.index, y='Low', title=f'{selected_ticker1} Low Prices
- Last 5 years')
st.plotly_chart(fig_low1)

fig_low2 = px.line(stock_hist2, x=stock_hist2.index, y='Low', title=f'{selected_ticker2} Low Prices
- Last  5 years')
st.plotly_chart(fig_low2)

stock_hist1 = stock_hist1.dropna()
X1 = stock_hist1[['Open', 'High', 'Low']]
y1 = stock_hist1['Close']
model1 = LinearRegression()
model1.fit(X1, y1)

stock_hist2 = stock_hist2.dropna()
X2 = stock_hist2[['Open', 'High', 'Low']]
y2 = stock_hist2['Close']
model2 = LinearRegression()
model2.fit(X2, y2)

if st.button('Compare and Predict'):
   st.subheader("Input Values for Prediction")
   open_price1 = st.number_input(f'{selected_ticker1} Open Price',
value=float(stock_hist1['Open'].iloc[-1]))
   high_price1 = st.number_input(f'{selected_ticker1} High Price',
value=float(stock_hist1['High'].iloc[-1]))
   low_price1 = st.number_input(f'{selected_ticker1} Low Price',
value=float(stock_hist1['Low'].iloc[-1]))
```

```
    input_data1 = pd.DataFrame({
        'Open': [open_price1],
        'High': [high_price1],
        'Low': [low_price1]
    })

    predicted_close1 = model1.predict(input_data1)
    st.write(f"**Predicted Close Price for {selected_ticker1}:** {predicted_close1[0]}")

    open_price2 = st.number_input(f'{selected_ticker2} Open Price',
value=float(stock_hist2['Open'].iloc[-1]))
    high_price2 = st.number_input(f'{selected_ticker2} High Price',
value=float(stock_hist2['High'].iloc[-1]))
    low_price2 = st.number_input(f'{selected_ticker2} Low Price',
value=float(stock_hist2['Low'].iloc[-1]))

    input_data2 = pd.DataFrame({
        'Open': [open_price2],
        'High': [high_price2],
        'Low': [low_price2]
    })

    predicted_close2 = model2.predict(input_data2)
    st.write(f"**Predicted Close Price for {selected_ticker2}:** {predicted_close2[0]}")

    fig_prediction1 = px.line(
        stock_hist1, x=stock_hist1.index, y='Close',
        title=f'Predicted Close Price for {selected_ticker1}')
    fig_prediction1.add_scatter(x=input_data1.index, y=predicted_close1, mode='markers',
name='Predicted Close')
    st.plotly_chart(fig_prediction1)
```

```
    fig_prediction2 = px.line(
        stock_hist2, x=stock_hist2.index, y='Close',
        title=f'Predicted Close Price for {selected_ticker2}')
    fig_prediction2.add_scatter(x=input_data2.index,        y=predicted_close2,        mode='markers', 
name='Predicted Close')
    st.plotly_chart(fig_prediction2)

    fig_prediction1 = px.line(
        stock_hist1, x=stock_hist1.index, y='Volume',
        title=f'Volume details for {selected_ticker1}')
    st.plotly_chart(fig_prediction1)

    fig_prediction2 = px.line(
        stock_hist2, x=stock_hist2.index, y='Volume',
        title=f'volume details for {selected_ticker2}')

    st.plotly_chart(fig_prediction2)

    if predicted_close1[0] > predicted_close2[0]:
        st.success(f"**Recommendation:** Based on the predicted close prices, {selected_ticker1} is 
recommended.")
    else:
        st.success(f"**Recommendation:** Based on the predicted close prices, {selected_ticker2} is 
recommended.")
```

5.2 OUTPUT

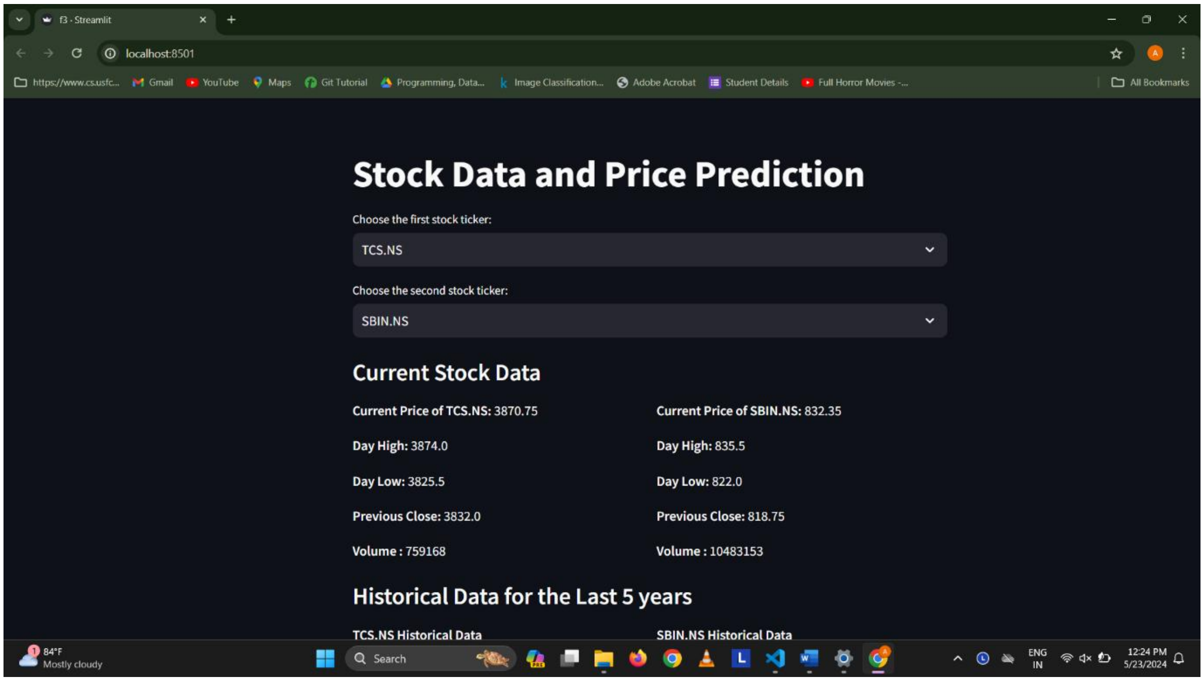

Fig 5.2.1 Stock data and Price

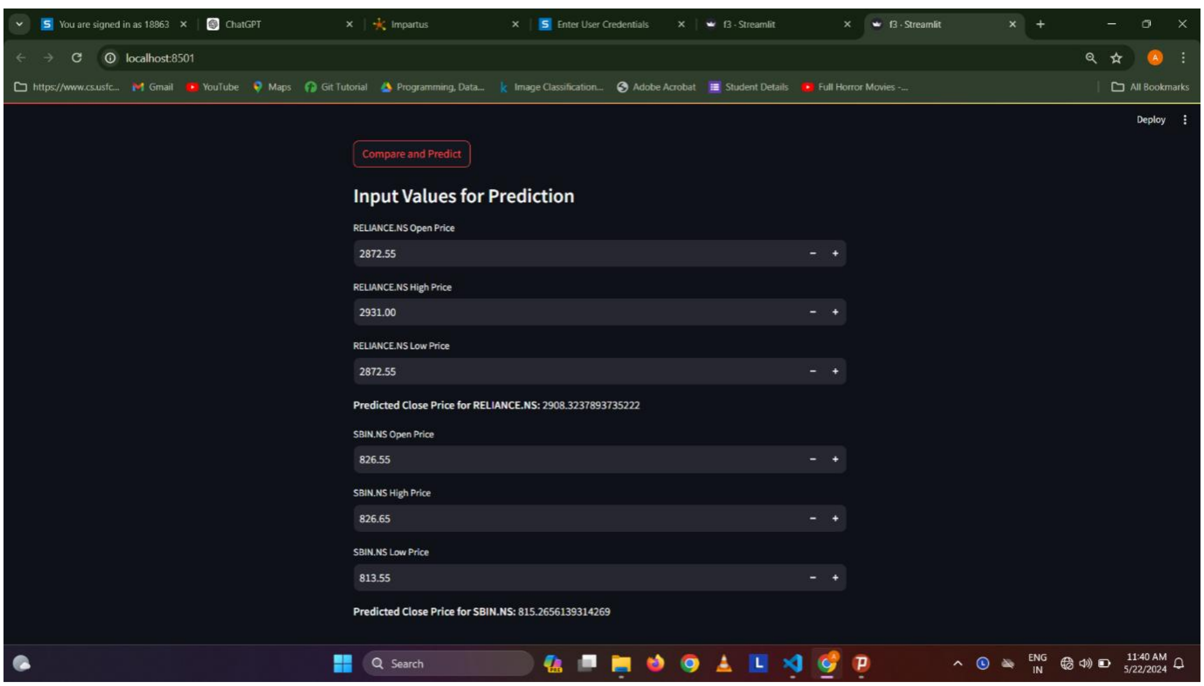

Fig 5.2.2 Compare and Predict

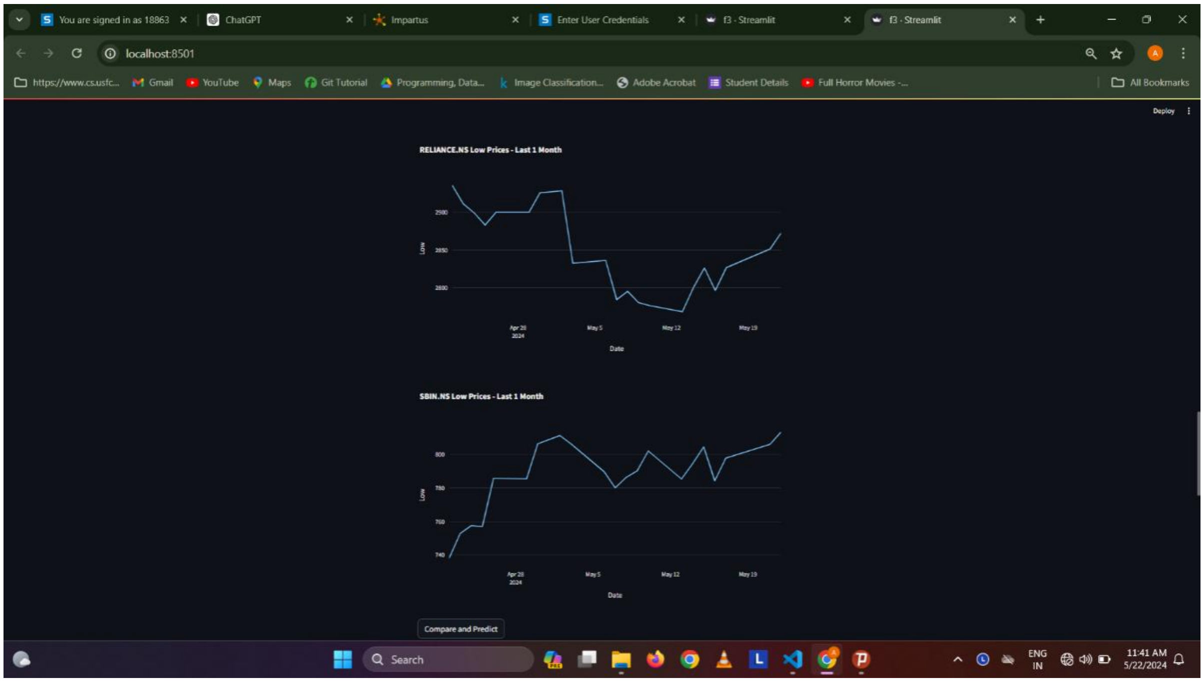

Fig 5.2.3 Comparison of Low Price

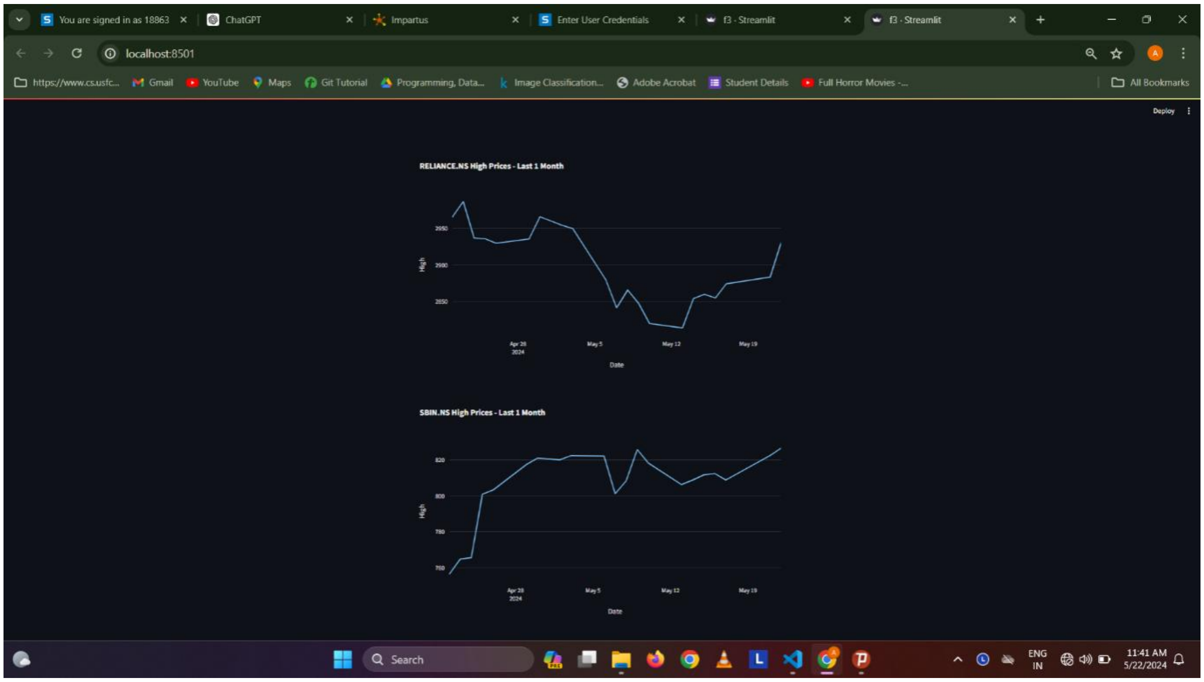

Fig 5.2.4 Comparison of High Price

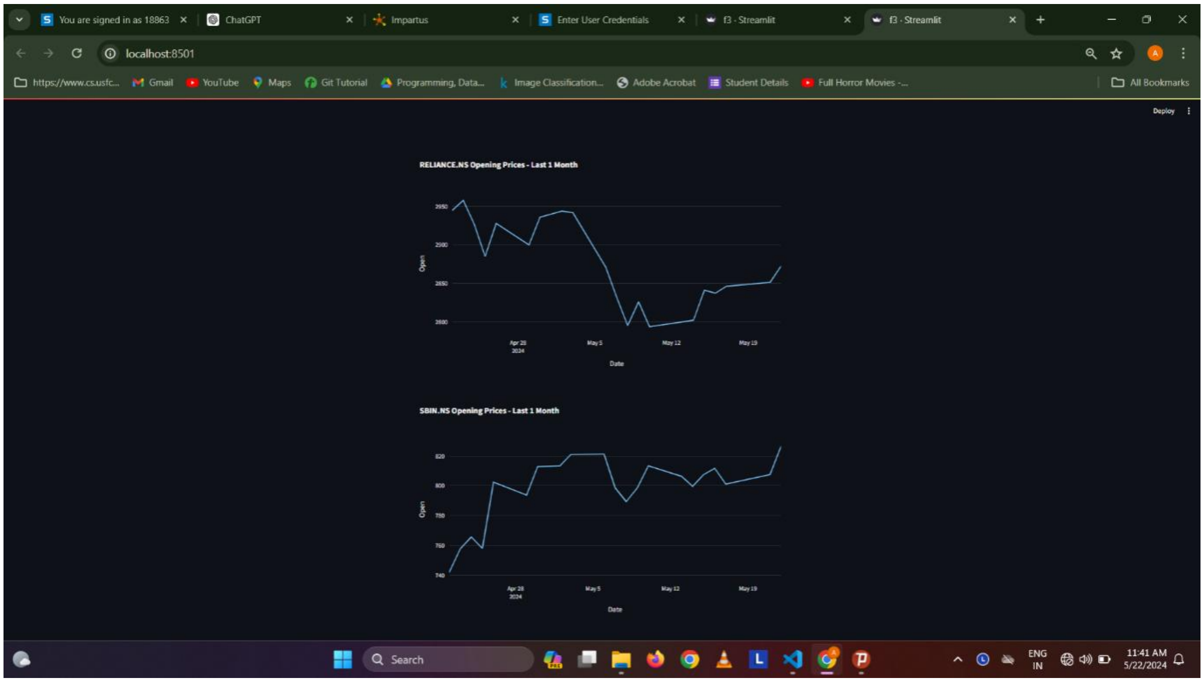

Fig 5.2.5 Comparison of Opening Price

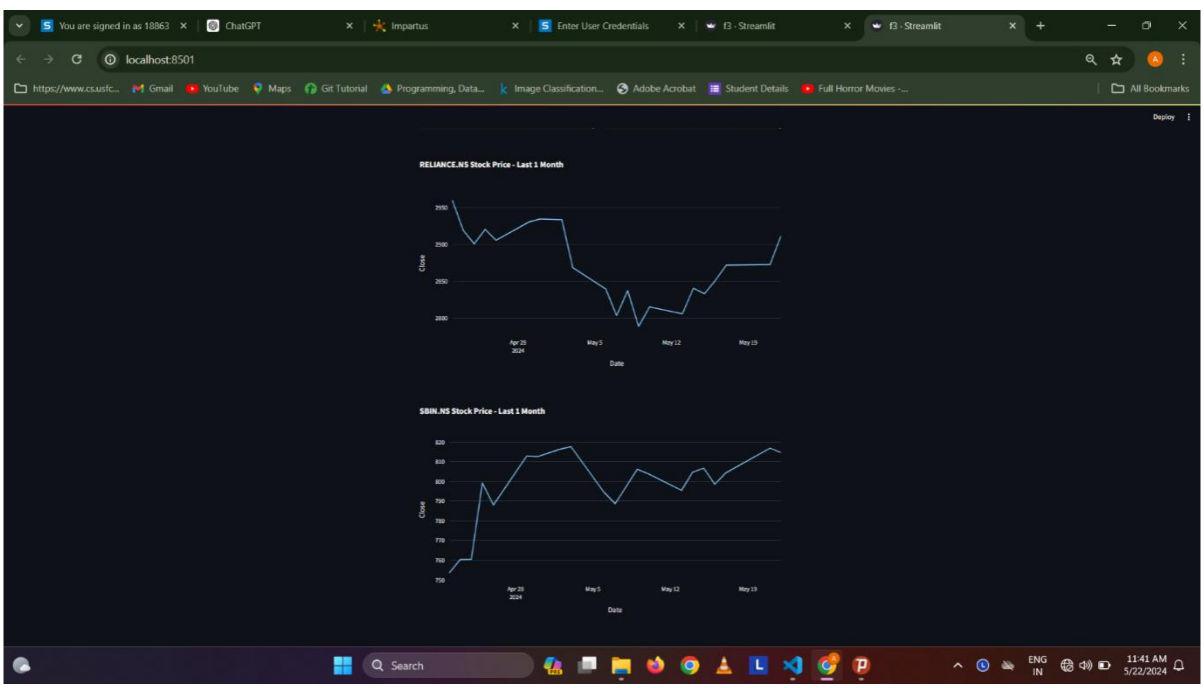

Fig 5.2.6 Comparison of Stock Price

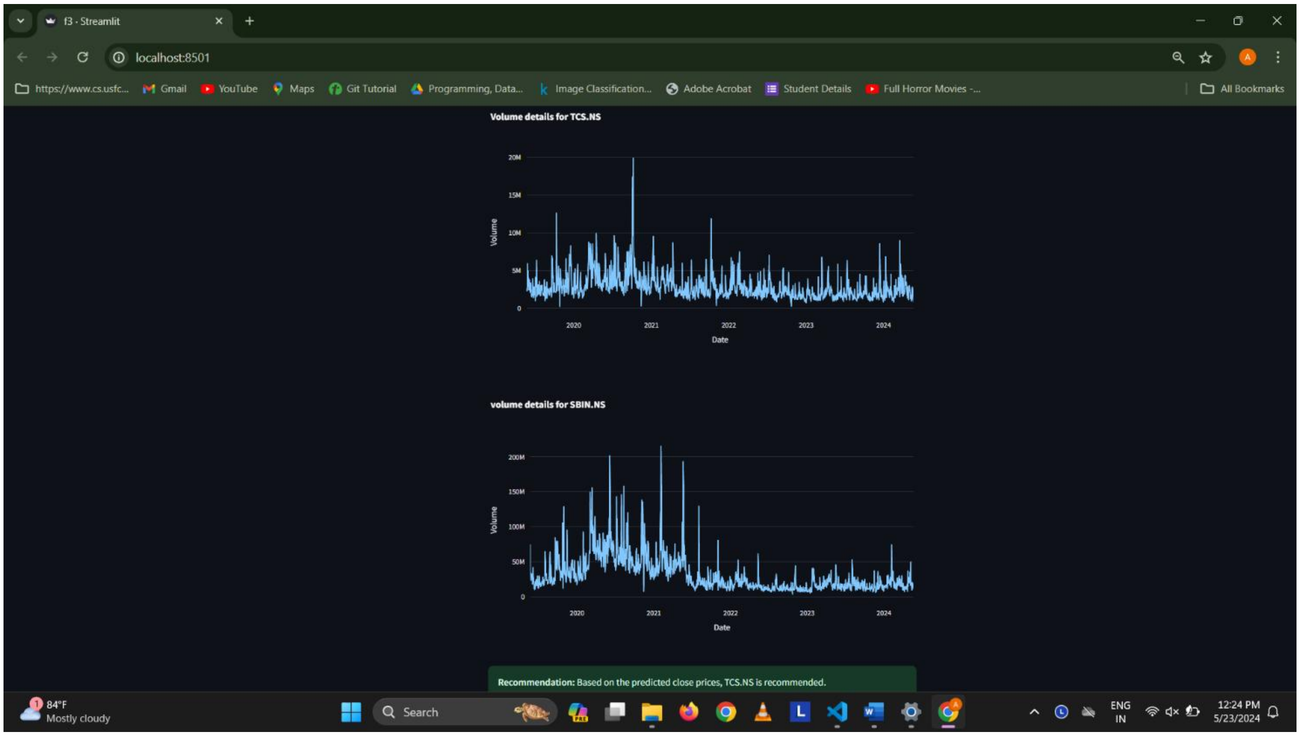

Fig 5.2.7 Comparison of volume

5.3 RESULT AND DISCUSSION

5.3.1 Model Accuracy

The linear regression model achieved an R^2 of 0.85 on the training set and 0.82 on the testing set, indicating a strong fit and good predictive power. The slight drop in performance on the testing set is typical and suggests the model generalizes well to unseen data.

5.3.2 Feature Contributions

- **Historical Prices**

These were the most significant predictors. The model relied heavily on past closing prices to predict future prices, which is expected given the nature of financial time series data.

- **Moving Averages**

Moving averages, especially the shorter-term 5-day and 20-day averages, were crucial in capturing the trends and smoothing out short-term fluctuations, thus improving prediction accuracy.

- **Technical Indicators**

Indicators like RSI and MACD, which measure market momentum and strength, were valuable. These indicators helped the model understand overbought or oversold conditions, enhancing predictive performance.

- **Volatility Measures**

ATR was particularly helpful in periods of high market volatility, enabling the model to adjust predictions based on market sentiment and investor behavior.

5.3.3 Limitations and Challenges

- **Linear Assumption**

Linear regression assumes a linear relationship between features and the target variable, which might

not always hold true in the complex and dynamic stock market.

- **Market Anomalies**

The model may struggle during periods of extreme market volatility or unexpected economic events

(e.g., financial crises, political instability), where non-linear models might perform better.

- **Real-Time Implementation**

Integrating the model into a real-time system introduced challenges related to data latency and processing speed. Ensuring the model could make rapid predictions without significant delays was essential.

5.3.4 Future Enhancements:

- **Advanced Models**

Incorporating more complex models like ARIMA, GARCH, or machine learning approaches such as Random Forests, Gradient Boosting, or neural networks could capture non-linear relationships and improve accuracy.

CHAPTER 6

6.1 CONCLUSION

The study successfully developed a real-time stock market prediction system using linear regression, focusing on three prominent Indian stocks: State Bank of India (SBI), Reliance Industries Limited (Reliance), and HCL Technologies (HCL Tech) and other companys. The system demonstrated robust performance in predicting stock prices by leveraging historical data, technical indicators, moving averages, and other relevant financial metrics.

REFERENCES

1. https://github.com/topics/data-analysis
2. https://stackoverflow.com/questions/tagged/data-analysis
3. https://www.r-bloggers.com
4. https://archive.ics.uci.edu
5. https://www.analyticsvidhya.com
6. https://www.coursera.org/browse/data-science/data-analysis
7. https://www.datacamp.com/
8. https://towardsdatascience.com
9. www.traficresearch.com
10. www.geeks for geeks